

Projet *Bitume Legends*



Rapport de Première Soutenance

CARRÉNIX

Anthony CARON – Melvyn DELAROQUE
Victorien CAMBOURIAN – Xavier DE PLACE

EPITA INFOSUP 2026
Année 2021 - 2022
btms.games

Table des matières

Introduction	2
1 Avancée du Projet	3
1.1 Multijoueur	3
1.2 Programme Beta (β)	3
1.3 Communication	3
1.4 Site Internet	3
1.5 Menu	3
2 Organisation interne chez CARRENIX	5
2.1 Organisation Pratique	5
2.2 Répartition des Tâches	5
2.2.1 Melvyn	5
2.2.2 Victorien	5
2.2.3 Anthony	6
2.2.4 Xavier	6
3 Ressources Utilisées	8
3.1 Collaboration	8
3.2 Communication	8
3.3 3D	8
3.4 Musique	9
3.5 Site Internet	9
3.6 IDE	10
4 Ressenti de chacun	11
4.1 Anthony	11
4.2 Melvyn	11
4.3 Victorien	11
4.4 Xavier	12
5 Objectifs pour la suite	13
5.1 Finir le menu	13
5.2 Finir le site Web	13
5.3 Continuer les musiques	13
5.4 Implémentation de l'Intelligence Artificielle	13
5.5 Gameplay	14
Conclusion	15
Annexes	16

Introduction

Nous sommes le studio CARRENIX qui développe *Bitume Legends*, un jeu de course automobile 3D réaliste en *low-poly*. Après avoir commencé le projet début janvier, nous publions notre premier rapport. Il permet de faire le point sur ce que nous avons fait, les objectifs atteints, les priorités ainsi que les ressources utilisés durant ce projet. Dans ce rapport, nous allons aborder l'avancée de notre projet, l'organisation de notre groupe, les ressources et créations de chacun des membres du groupe ainsi que leurs valeurs ajoutées, le ressenti des membres durant cette première période et nos objectifs.

1 Avancée du Projet

1.1 Multijoueur

La base du multijoueur a été implémentée, non sans peines. Nous pouvons nous connecter à 4 personnes (nous avons choisi ce nombre pour des questions de simplicité) et commencer une partie. Il y a une interface de connection, avec différentes options (détaillées plus tard dans ce document) puis le système de *spawn* (appartition). Une musique de jeu a également été créée pour le multijoueur.

1.2 Programme Beta (β)

Nous avons créé un questionnaire sur *Google Docs* afin que les premières personnes testant le jeu puisse nous signaler les bugs qu'ils ont rencontrés et nous les décrire le plus précisément possible dans le but d'être corrigé rapidement. Pour cela, ce questionnaire possède pour trois catégories. Tout d'abord, une question sur le mode de jeu où le bug s'est produit. Puis une autre sur la nature du bug et enfin un paragraphe réservé pour nous décrire avec le plus de détails possible ce bug. Toutefois, si l'utilisateur trouve compliqué de décrire son bug, des liens pour nous contacter sur *Discord* ou *GitHub* sont indiqués.

1.3 Communication

Au sujet de la communication, nous avons décidé de faire une publication sur *Instagram*¹ pour tenir informés nos potentiels premiers joueurs. Dans cette publication se trouve le lien du site Internet où l'utilisateur pourra consulter se familiariser avec le menu de notre jeu, puisque le site reprend l'esthétique de ce dernier. Enfin l'onglet de la réalisation et celui de la présentation donnent l'occasion de nous connaître plus, de même pour le projet. Ils pourront enfin nous rejoindre sur *Discord* pour que nous puissions les inscrire dans le programme de tests.

1.4 Site Internet

TODO ANTHONY

1.5 Menu

Le menu doit permettre d'accéder aux différents modes de jeux et fonctionnalités du jeu. Il doit servir de hub pour accéder au mode multijoueur, contre la montre et contre l'Intelligence Artificielle. Il doit également permettre d'accéder

1. [instagram.com/bitumelegends](https://www.instagram.com/bitumelegends)

au garage contenant les voitures du joueur. Un affichage du niveau du joueur, un bouton pour accéder à la remontée de bug et un menu d'options sont également dans le menu du jeu. Au début du projet, le menu était basique. Il servait simplement à pouvoir accéder aux différentes modes de jeux. Suite à cela, Melvyn et Victorien ont commencé à travailler sur un design plus poussé que celui de base de *Unity*, en adoptant notamment un thème de couleurs basé sur le logo du jeu. Tout d'abord, une ébauche du design a été réalisée sur Canva, dans le but de partager des idées sur le look final du menu. Ensuite nous avons implémenté le design dans le jeu, en insérant un fond et des espaces pour les futures fonctionnalités, en implémentant les boutons et en changeant leur design. Nous avons également implémenté le code couleur du design. Enfin, nous avons ajouté des effets sonores et de couleurs aux boutons lorsqu'ils sont cliqués ou lorsque la souris passe dessus. Une musique tourne également en fond dans le menu.

2 Organisation interne chez CARRENIX

2.1 Organisation Pratique

Nous avons mis en place une organisation particulière entre nous, car au début nous ne savions pas spécialement par où commencer. Chaque semaine, nous nous sommes réunis pour définir des objectifs pour chacun, à faire durant la semaine suivante. Cela nous a permis d'avoir des buts concrets sur le court terme et d'avancer plus efficacement.

2.2 Répartition des Tâches

Nous nous sommes basés sur ce que nous avions annoncé dans notre Cahier des Charges : Melvyn s'est occupé des musiques d'ambiance, du *Sound Design* de notre jeu et du menu principal. Victorien a créé le premier circuit du jeu, il a implémenté le menu principal et a modélisé sur *Blender* une Formule 1. Anthony a géré le site Internet, les réseaux sociaux et le programme β . Enfin, Xavier a implémenté le mode multijoueur, et le cœur du moteur de jeu, à savoir le système de direction des voitures.

2.2.1 Melvyn

Étant responsable de la musique, j'ai composé la musique du jeu. J'ai utilisé le logiciel de MAO (Musique Assistée par Ordinateur) *FL Studio 20*. J'ai composé deux musiques, une musique pour le menu et une musique de course et j'ai demandé à un ami d'enregistrer sa voix pour la musique dans un studio d'enregistrement de son école. Le style choisi par mes pairs et moi-même au sein du groupe est la *Phonk*, style est souvent associé à la culture automobile *underground* japonaise, d'où le désir d'utiliser de la *Phonk* pour notre jeu. De plus, ce style est très énergique, assourdissant parfois et fait monter l'adrénaline dans le sang. Au début, j'ai hésité à faire de l'*Eurobeat*, un genre de *dance music* associé au milieu automobile, mais le groupe a tranché et nous avons choisi un genre qui nous est à la fois plus familier et bien plus facile à produire comparé à de la musique électronique. De plus je suis responsable du design et j'ai travaillé sur le design des menus, ainsi que sur l'identité graphique du jeu et l'identité sonore de *Bitume Legends*.

2.2.2 Victorien

Étant donné que nous créons un jeu de voiture, il était bienvenu de modéliser au moins une voiture par nos propres soins. Ayant des connaissances dans *Blender*, un logiciel de modélisation 3D, je m'en suis chargé. Nous avons convenu d'une Formule 1, voiture considéré comme le "Graal" du jeu. Les formes de la voiture

ont toutes été créées à partir de formes basiques, telles que des rectangles, des cylindres, des sphères ou encore des cubes. Ensuite, il ne restait qu'à jouer avec les formes : les étirer, redimensionner une face pour que les formes correspondent avec une Formule 1. Vu que le design est en *low-poly*, nous avons rajouter des faces à certaines formes. Par exemple, avec 26 rectangles verticaux collés les uns aux autres, on peut former un cylindre. Pour essayer d'être réaliste, nous avons regarder une vidéo présentant une Formule 1 avec ses détails telles que le volant, les suspensions, les ailerons et les moustaches de la voiture.

2.2.3 Anthony

Pour le site Web, nous avons décidé de reprendre l'esthétique du menu de notre jeu vidéo. Pour cela, nous avons divisé le site web en quatre parties, accessibles par quatre boutons amenant aux quatre sous-pages de notre site. La première est la présentation du projet, dans laquelle nous avons écrit une courte introduction, l'historique et les membres de l'équipe. La seconde concerne la réalisation du projet, qui contient la chronologie de la réalisation, les problèmes rencontrés et solutions envisagées pour contrer ceux-ci. Ensuite, une troisième page nous offre des liens de téléchargement du jeu pour Windows et macOS, et le projet en version complète ou allégée. Enfin, la dernière page permet l'accès aux différentes ressources (liens des images, sons, logiciels...). C'est une sorte de bibliothèque des sources de notre projet. Ainsi, il me manquera plus qu'à faire fonctionner certains boutons et liens et rendre le site responsive, c'est à dire un site qui s'adapte en fonction de l'appareil depuis lequel on le consulte, et régler quelques bugs d'affichages pour finaliser le site Web.

2.2.4 Xavier

Pour faire tourner un jeu, il faut que chaque élément soit relié avec les autres grâce à des liens dans le code. Dans notre cas, cela peut être appliqué aux voitures, qui doivent être des objets controllables et visbles. Pour cela, nous avons intégré les voitures dans le moteur de jeu *Unity* en rajoutant des composants tel que le *Rigidbody*, le centre de gravité, les *Colliders*, et les *Wheel Colliders*. Le premier est un élément permettant de simuler les propriétés physiques de la voiture, comme la masse ou la détection de collisions. Le second est un élément du *Rigidbody*, qui est sous la forme d'un point précis mis sur la voiture, et qui est le point d'application de la gravité. C'est un élément pratique, qui permet de définir comment la voiture va réagir dans les virages, ou comment celle-ci va *drifter*. Le troisième élément à intégrer est en plusieurs parties, chacune pour un morceau de la voiture (capot, toit, ailes, etc.). Ce sont des zones virtuelles autour de la voiture qui permettent de la rendre solide, c'est à dire qu'on ne peut pas traverser, et qui ne peut pas traverser non plus les murs et autres éléments. Et enfin, les *Wheel Colliders* sont des

éléments appliqués sur chaque roue et qui jouent le rôle de moteur. Pour les roues arrières, ils transmettent l'action du joueur si il appuie sur les touches pour avancer et reculer, en entrainant les roues (et donc la voiture) vers la bonne direction. Et pour les roues avant, ils s'occupent de faire tourner la voiture en fonction des actions du joueur. Après cela, nous avons intégré le mode de jeu mutlijoueur. Nous avons décidé d'utiliser *Photon Cloud*, une solution gratuite, *open source*, et relativement "simple" à implémenter. Il y a une bonne documentation sur Internet, et de nombreux tutoriaux sur YouTube. Nous avons commencé par créer une scène pour la connection, qui, grâce à une interface graphique, permet de choisir un nom. Ensuite, nous pouvons créer une *room* (instance de multijoueur), d'en chercher une déjà ouverte ou encore d'en joindre une directement si nous connaissons son nom. Enfin, une fois que la *room* est créée, le jeu lance une scène comportant un circuit, et le jeu peut commencer.

3 Ressources Utilisées

3.1 Collaboration

La création de jeux en solo ne demande pas de partage de données, contrairement à celle en équipe. Nous avons besoin d'un endroit de collaboration où nous pourrions nous échanger les fichiers relatifs au jeu. Nous avons donc créé un ensemble de *repositories* (ou *repo*) sur *Github*², répartis dans une organisation, chacun pour un usage bien spécifique. Le premier est donc le *repo* de notre jeu, nommé *game*, qui contient le projet au format *.unity* et des dossiers contenant les différents assets et autres ressources, nécessaires à l'exécution du jeu. Le second est un *repo* consacré exclusivement aux différents rapports que nous devons fournir. Il est composé à 99% de *.TEX* et 1% de *.pdf*. Enfin, le dernier *repo* est celui dédié à notre site Internet, qui possède une double fonction : il nous permet de collaborer sur le site mais aussi de l'héberger grâce à *GitHub Pages*.

Ainsi, nous pouvons toujours être à jour sur la bonne version du jeu, du site ou des rapports, tout en étant géographiquement à distance les uns des autres.

3.2 Communication

Pour communiquer entre nous et avec notre équipe de β -testeurs, nous avons créé un serveur Discord³ découpé en multiples *channels*, ayant chacun une mission précise pour ne pas mélanger les informations. Ce serveur est aussi le lieu de nos réunions hebdomadaires (ou plus fréquemment en cas de soutenance). De plus, depuis le site web, nous avons mis un lien vers un questionnaire de remontée de bugs, via un *Google Docs*. Nous avons également mis en place un calendrier collectif lors de la dernière semaine avant la soutenance. Cela nous a offert une vision plus claire des tâches individuelles et de les découper en plages horaires afin de respecter les *deadlines* et de ne pas avoir d'erreurs de versions entre nous.

3.3 3D

Comme moteur de jeu, nous avons utilisé *Unity*. Comme Xavier utilise un Mac et que le reste du groupe est sous Windows, nous utilisons la version 2021.2.7f1 qui fonctionne sur les deux OS. Nous avons décidé de geler la version pour limiter au maximum les problèmes d'incompatibilité entre nous. Nous avons choisi *Unity* pour sa simplicité de prise en main et une fonctionnalité très utile : l'*Asset Store*. C'est une plateforme où nous pouvons acheter ou utiliser gratuitement des ressources telles que des bâtiments, des voitures ainsi que des personnages. En plus

2. [github.com/Bitume – Legends – Crew](https://github.com/Bitume-Legends-Crew)

3. discord.gg/5NR43GHUBD

de l'utilisation de l'*Asset Store* pour notre première voiture, nous avons aussi utilisé *Blender*, logiciel de modélisation 3D. Il nous a permis entre autres de créer une Formule 1 de 2021 qui sera utilisée et implémentée plus tard dans le jeu.

3.4 Musique

La musique a été composée sur *FL Studio 20* logiciel de MAO, dans un genre *Phonk*. La *Phonk* est un genre de *Trap* rapide (tempo de 160 battements par minute) et très énergique. Ce genre est connu pour ses samples (extraits musicaux) de voix tirés de morceaux de rap de Memphis et pour ses percussions. En effet, les instruments de ce genre sont presque tous des percussions. Celle qui caractérise la *Phonk* est la *cowbell* (cloche autour du cou des vaches) venant de la *Roland TR-808* (boîte à rythme mémorable de la culture hip-hop). Un ensemble de percussions cohérent, rapide et laissant une place majeure aux *kicks* (grosses caisses) et *hi-hat* (cymbales *Charlestone*) est primordial. Enfin, l'identité sonore du genre veut que le son soit un peu saturé, trop compressé. Cela correspond au mouvement *low fidelity* (ou *lofi*), courant de pensée musicale préférant un rendu imparfait à une exactitude que donne normalement un ordinateur, quitte à artificiellement dégrader la qualité sonore.

Le processus de composition a commencé par l'écriture de la mélodie principale. Ensuite vient le travail des percussions (cruciales pour la *Phonk*) représentant 50% du travail total. Ensuite nous avons travaillé sur le *mixing* et le *leveling*, procédés de travail du volume, de la stéréo et d'effets de spacialisation du son. Cette étape représente la deuxième partie la plus importante de la composition. C'est à cette étape que l'on va s'assurer de donner une bonne qualité sonore à la musique, que l'ensemble des sons s'accorde et soit cohérent. Pour faire une analogie, le contrôle des niveaux (*leveling*) est équivalent à ce que fait un chef d'orchestre. Enfin, l'arrangement et le *mastering* permettant, pour le premier, une structure cohérente de la musique, et pour le second, de légères retouches sonores faisant toute la différence. Cette dernière partie du processus est la plus longue mais celle sur laquelle on peut exprimer sa créativité, car c'est ici que l'on peut l'on peut travailler sur des variations, des transitions et ajouter du punch.

3.5 Site Internet

Pour réaliser le site Internet, nous avons choisi d'utiliser *Bootstrap Studio 5* afin de limiter le développement en HTML et CSS. *Bootstrap Studio* est une application de conception de site Web très puissante, nous pouvons choisir des structures déjà prédéfinies comme des colonnes et des lignes puis y ajouter ce que l'on souhaite, notamment des boutons, des images, des liens, du texte... Enfin,

Bootstrap Studio nous propose de personnaliser ces derniers éléments en modifiant leurs styles et leur place afin de limiter l'usage du CSS. Nous sommes quand même passé par le HTML pour customiser plus profondément certains éléments de notre site, comme par exemple les boutons. C'est alors qu'intervient l'éditeur de texte *Visual Studio Code* dans lequel nous avons exporté le code généré par Bootstrap et intégré les propriétés manquantes à chaque bouton. Pour mettre en ligne le site une fois terminé, nous avons utilisé la solution *GitHub Pages* pour l'hébergement. Nous avons raccordé le nom de domaine à notre *repository* puis nous avons configuré les *DNS* (le système qui permet à un site web d'être visible sur Internet) pour que tout fonctionne bien. Le design du site est repris du menu du jeu nous permettant de rester dans l'identité graphique du projet.

3.6 IDE

Pour écrire notre code en C#, nous utilisons l'IDE *Rider* de *JetBrains*. Il possède une bonne intégration de Unity, et nous y sommes bien habitué, c'est celui que nous utilisons au quotidien pour nos TPs de programmation. Pour faire certains tests, très précis et qui ne nécessitent pas de beaucoup de ressources, nous utilisons *Vim* directement dans notre terminal. Pour collaborer sur le rapport et les autres documents en \LaTeX pour éviter de désigner un "esclave \LaTeX ", nous utilisons le site *Overleaf*⁴. À la manière d'un document en ligne, comme un *Google Docs* ou autre, nous pouvons écrire en simultané et compléter à quatre cerveaux les documents demandés.

4. overleaf.com

4 Ressenti de chacun

4.1 Anthony

L'organisation du travail au sein du groupe était plutôt bonne. Dès le début, nous avons créé un repo GitHub afin de partager notre travail effectué ainsi qu'un Discord pour se mettre d'accord sur les différentes deadlines à propos du travail à effectué chaque semaine. Ainsi nous effectuons toutes les semaines, des réunions en vocal, dans le but d'expliquer aux autres membres du groupe ce que l'on a fait dans la semaine. A propos du site web, j'ai eu du mal à manipuler Bootstrap Studio du fait du manque de tutoriaux pour apprendre à utiliser Bootstrap Studio sans de développement en HTML. J'ai donc dû apprendre comment fonctionne ce logiciel et après avoir réalisé que le principe était de mettre tous les éléments liés dans une seule colonne, Bootstrap est devenu toute de suite très facile et m'a permis de réaliser rapidement le site web.

4.2 Melvyn

Entre la première présentation du projet et le rapport de soutenance j'ai découvert les différentes difficultés liées à la réalisation d'un projet. J'ai eu du mal avec l'investissement personnel et la gestion du temps durant ce mois et demi. Je compte améliorer ma manière de travailler et changer mon processus de composition afin de pouvoir être plus investi dans le projet et produire plus de contenu. Je suis heureux de pouvoir travailler sur un style de jeu que j'aime. Je suis également ravi de pouvoir à nouveau travailler sur des projets musicaux et de pouvoir mettre ma passion au service du projet. Un aspect intéressant du projet a également été de pouvoir implémenter nos idées en utilisant des méthodes qui m'étaient dès lors inconnues. Ce fut un challenge mais j'ai trouvé cela amusant et suis prêt pour la suite.

4.3 Victorien

Quelques années auparavant, j'avais déjà créé un jeux-vidéo sur un autre moteur de jeux, Unreal Engine de Epic Games. Je n'avait cependant pas eu à coder car il suffisait de créer des Blueprints (éléments de code préfabriqués qu'il faut relier entre eux). De plus, c'était un jeu développé en solo, pas en équipe. Je ne savais pas spécialement par quoi commencer au vu de ce que l'on avait prévu. Après avoir fixé des objectifs hebdomadaires grâce aux réunions, les choses étaient cadrées et j'ai pu être plus efficace sur le développement du jeu. Il a fallu également faire attention à notre emploi du temps. Bien qu'il y eût les vacances pour avancer, il y avait à la rentrée les Midterms et 5 jours plus tard première soutenance. Nous devions être organisé et voir plus loin que la semaine suivante.

4.4 Xavier

Todo

5 Objectifs pour la suite

5.1 Finir le menu

Le menu principal du jeu est presque fini, il nous manque un onglet pour les paramètres du jeu et il faut que nous relions les autres fonctions du jeu tel que le mode *Timer* ou le mode *vs Intelligence Artificielle* à des nouvelles scènes pour y accéder. Ensuite, nous devons mettre au point le menu *Player*, en rajoutant la possibilité de voir son niveau, de modifier son nom, etc. Un sous-menu *Options* est aussi à implémenter.

Les options comporteront :

- modification le volume de la musique
- modification le volume des effets sonores
- changement de la musique du menu
- changement du nom du joueur

5.2 Finir le site Web

Le site web est bien avancé, mais il nous manque certains éléments comme la fin de la chronologie, la page de téléchargement avec les liens, ou encore quelques bugs concernant l’affichage tel que l’affichage du site web sur téléphone, ou alors avec une résolution en écran fenêtré sur ordinateur.

5.3 Continuer les musiques

Une de nos plus-values étant la musique, nous nous devons de développer cet aspect ainsi que la manière de les implémenter dans le jeu. C’est pourquoi nous comptons produire 3 musiques en plus pour le jeu : une autre musique de menu et deux autres de courses. Il serait dommage qu’un jeu de course énergique semble répétitif à cause de la musique. Ces musiques restaeront dans un thème *Phonk* mais plus inspirés d’artistes comme Freddi Dredd, h8.hood ou encore Ghostemane. Ces *soundtrack Phonk* auront alors plus une esthétique Métal ou *HorrorCore* issue du rap de Memphis. Cela impliquera alors un possible changement d’instrument pour la mélodie au lieu de la *cowbell*.

5.4 Implémentation de l’Intelligence Artificielle

Nous avons déjà implémenté le mode multijoueur, mais ce n’est pas le seul mode prévu. Le suivant est le mode contre une Intelligence Artificielle, mode de jeu qui va permettre au joueur de jouer en "multijoueur" sans forcément avoir un joueur physique en face de lui. Ce mode risque d’être un peu long à développer,

il demande beaucoup de précision dans le code, et risque de générer beaucoup d'erreurs.

5.5 Gameplay

Par *Gameplay*, nous entendons la gestion de l'expérience et des niveaux, qui sera gagné en fonction de la place dans le classement à la fin de la course. Mais aussi ce qui se rapporte au choix d'une voiture avec une rotation des circuits pour commencer une partie. La personnalisation des voitures est également incluse grâce à l'implémentation d'un garage, où l'utilisateur pourra choisir sa voiture depuis le menu.

Conclusion

En conclusion, malgré / après vous avoir énoncé nos objectifs, d'ici la prochaine soutenance ils ne seront pas tous terminés. Nous avons prévu d'avoir terminé les différents modes de jeu, dont l'IA. De plus, nous devons avoir les différents circuits, Nous serons ravis de vous revoir le **PROCHAINE DATE DE SOUTENANCE**

Références

- blender.org
- bootstrapstudio.io
- discord.com
- unity.com
- photonengine.com/pun
- overleaf.com
- jetbrains.com/rider
- assetstore.unity.com
- [image – line.com/fl – studio](https://image-line.com/fl-studio)

Made with ♥ by CARRENIX on L^AT_EX.

© 2021-2022 *Bitume Legends*
btms.games

Annexes

Table des figures

1	Arborescence du menu	17
	il faut encore caler :	
	— F1	
	— capture du site	
	— screen du bugreport	

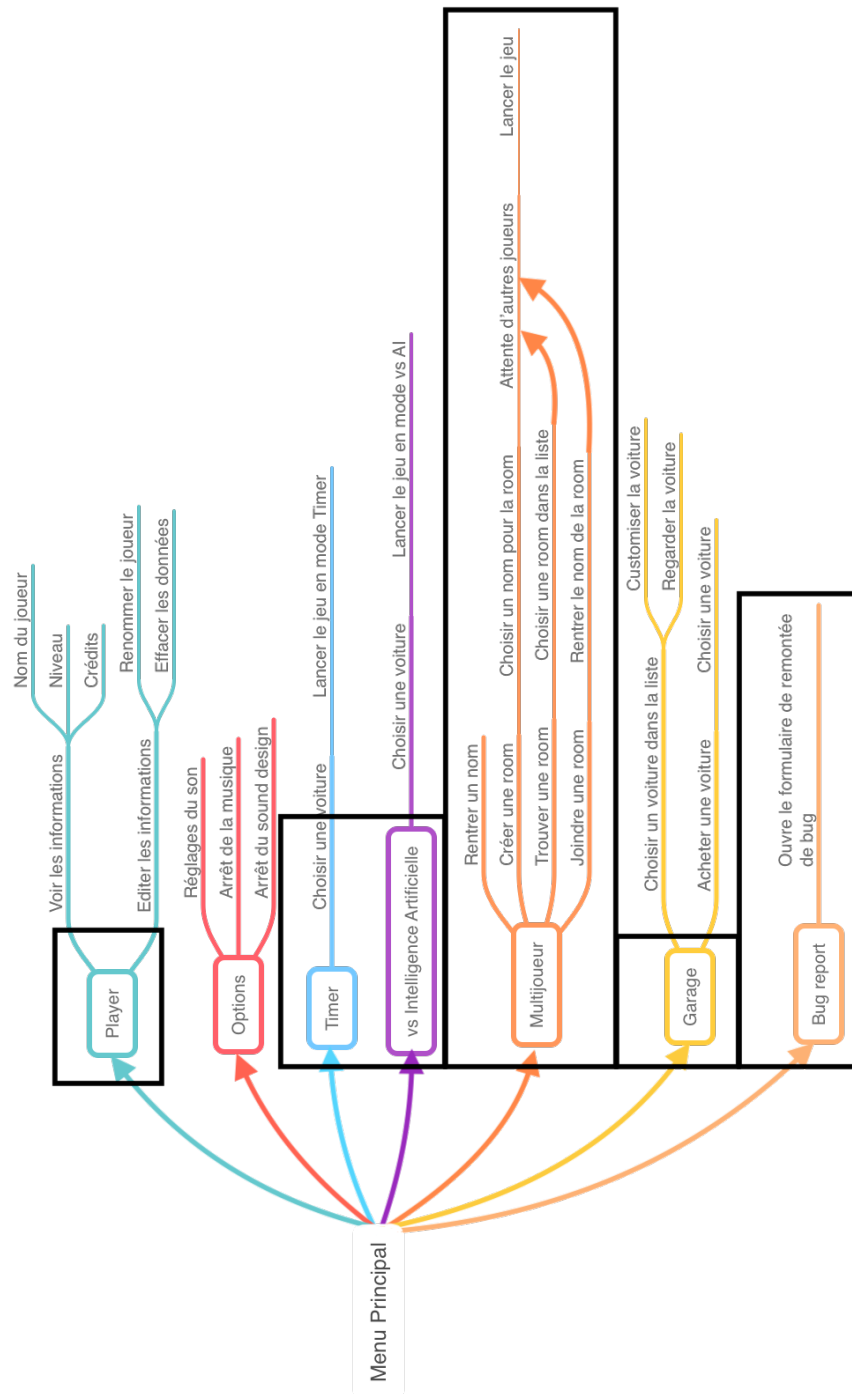


FIGURE 1 – Arborescence du menu
Ce qui est entouré est déjà implémenté, le reste suivra.